

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開 2001-357406

(P 2001-357406 A)

(43) 公開日 平成13年12月26日 (2001. 12. 26)

(51) Int. Cl. 7	識別記号	F I	テーマコード (参考)
G 0 6 T	7/60 2 5 0	G 0 6 T	7/60 2 5 0 Z 5B057
	3/00 5 0 0		3/00 5 0 0 Z 5L096
	7/00 1 0 0		7/00 1 0 0 Z

審査請求 未請求 請求項の数 9

OL

(全 17 頁)

(21) 出願番号 特願2000-178823 (P2000-178823)

(22) 出願日 平成12年6月14日 (2000. 6. 14)

(71) 出願人 000006079

ミノルタ株式会社

大阪府大阪市中央区安土町二丁目3番13号

大阪国際ビル

(72) 発明者 田中 芳則

大阪市中央区安土町二丁目3番13号 大阪

国際ビル ミノルタ株式会社内

(74) 代理人 100072349

弁理士 八田 幹雄 (外4名)

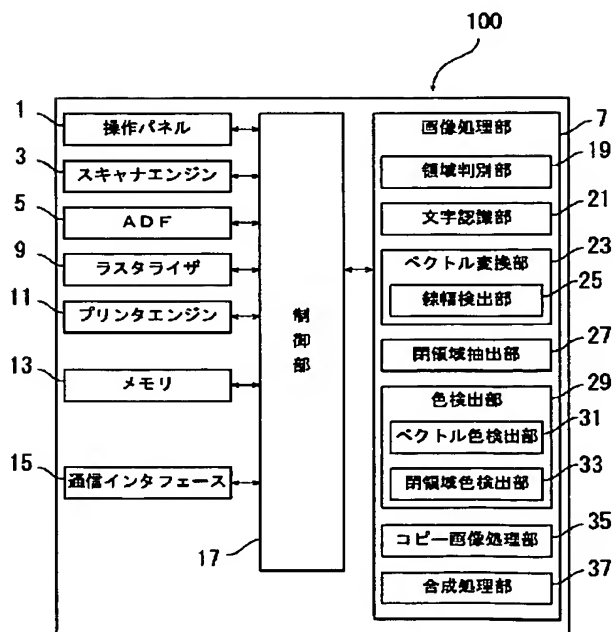
最終頁に続く

(54) 【発明の名称】 画像処理装置

(57) 【要約】

【課題】 図形の色を再現でき、ベクトル変換後のデータ量をより一層低減でき、枠線付き閉領域の拡大／縮小による線幅の変化を防止でき、データ再利用時の使い勝手を向上できる画像処理技術を提供する。

【解決手段】 ベクトル変換部 23 は、図形領域に含まれる線図形（線分や曲線）を認識して、線幅の情報を保持ベクトルデータに変換する。このとき、線図形の線幅は、線幅検出部 25 で検出される。閉領域抽出部 27 は、ベクトルデータに基づいて閉領域を抽出する。色検出部 29 は、ベクトルデータに付加される色情報（ベクトルの色、閉領域の色）を検出する。このとき、ベクトルの色は、ベクトル色検出部 31 で検出され、閉領域の色は、線幅の情報を考慮して、閉領域色検出部 33 で検出される。



【特許請求の範囲】

【請求項 1】 画像処理装置において、
図形をベクトルデータに変換するベクトル変換手段と、
ベクトルデータに変換された線図形の線幅を検出する線
幅検出手段と、
変換されたベクトルデータに基づいて閉領域を抽出する
閉領域抽出手段と、
検出された線幅の情報に応じて、抽出された閉領域の内
部の色を検出する閉領域色検出手段と、
を有することを特徴とする画像処理装置。

【請求項 2】 検出された線幅の情報は、対応する線図
形のベクトルデータに付加されることを特徴とする請求
項 1 記載の画像処理装置。

【請求項 3】 検出された線幅が設定値以上であるか否
かを判断する判断手段をさらに有し、
前記ベクトル変換手段は、
検出された線幅が設定値以上である場合、対応する線図
形のエッジを検出し、検出したエッジをベクトルデータ
に変換することを特徴とする請求項 1 記載の画像処理装
置。

【請求項 4】 前記閉領域色検出手段は、
検出された線幅の情報に応じて、抽出された閉領域の内
部点であって当該閉領域を形成するベクトルデータに対
応する線図形よりも内側に存在する内部点を選択する選
択手段を含み、
選択された内部点の色情報に基づいて、抽出された閉領
域の内部の色を検出することを特徴とする請求項 1 記載
の画像処理装置。

【請求項 5】 ベクトルデータに変換された線図形の色
を検出する線色検出手段をさらに有することを特徴とす
る請求項 1 記載の画像処理装置。

【請求項 6】 検出された色の情報は、対応する線図形
のベクトルデータに付加されることを特徴とする請求項
5 記載の画像処理装置。

【請求項 7】 請求項 1 記載の画像処理装置は、画像読
取装置に搭載されていることを特徴とする。

【請求項 8】 画像処理方法において、
図形をベクトルデータに変換するステップと、
ベクトルデータに変換された線図形の線幅を検出するス
テップと、
変換されたベクトルデータに基づいて閉領域を抽出する
ステップと、
検出された線幅の情報に応じて、抽出された閉領域の内
部の色を検出するステップと、
を有することを特徴とする画像処理方法。

【請求項 9】 画像処理プログラムを記録したコンピュ
ータ読取可能な記録媒体であって、該画像処理プログラ
ムが、
図形をベクトルデータに変換するステップと、
ベクトルデータに変換された線図形の線幅を検出するス

テップと、
変換されたベクトルデータに基づいて閉領域を抽出する
ステップと、
検出された線幅の情報に応じて、抽出された閉領域の内
部の色を検出するステップと、
を有することを特徴とするコンピュータ読取可能な記録
媒体。

【発明の詳細な説明】

【0001】

10 【発明の属する技術分野】本発明は、図形画像を処理す
るための画像処理装置および方法ならびに画像処理プロ
グラムを記録したコンピュータ読取可能な記録媒体に関
する。

【0002】

【従来の技術】図形の拡大／縮小、情報圧縮などに利用
するため、スキャナなどから入力されたラスタデータを
ベクトルデータに変換すること（通常「ベクトル変換」
と呼ばれる）が行われている。すなわち、線図形を使っ
て図形を表現すると、拡大／縮小しても形が変わらず、
20 また、図形情報がコンパクトになり情報圧縮になるた
め、線図形を表現する方法として、図形（ラスタデー
タ）の輪郭線をベクトルデータに変換することが行われ
ている。

【0003】たとえば、特開平 6-195421 号公報
には、ラスタデータから色により輪郭線（エッジ）デー
タを抽出し、これをベクトルデータに変換し、座標変換
した後、変換された座標に基づいて輪郭線を描画し、各
輪郭線の色で輪郭線の内部を塗り潰すことが記載されて
いる。

【0004】

【発明が解決しようとする課題】しかしながら、このよ
うな従来のベクトル変換技術にあっては、エッジを検出
し、エッジデータをベクトルデータに変換するため、線
図形の線幅が 1 よりも大きい場合（細線化図形の線幅を
1 とする）には、次のような問題がある。

【0005】たとえば、図 16 に示すように、黒色の線
分を従来の方法でベクトル変換した場合、線分は、4 本
のベクトルで囲まれた閉領域で内部が黒色で塗り潰され
ているものとして認識される。つまり、この場合、1 本
40 の線分に対して 4 本のベクトル情報を持たなければなら
ない。

【0006】また、たとえば、図 17 に示すように、黒
色の枠線を持つ閉領域で内部が緑色で塗り潰されたもの
を従来の方法でベクトル変換した場合、かかる枠線付き
閉領域は、2 つの閉領域としてベクトル化され、外側の
大きい四角形は内部が黒色で塗り潰され、内側の小さい
四角形は内部が緑色で塗り潰されているものとして認識
される。つまり、この場合、1 つの閉領域に対して 2 つ
の閉領域情報を持たなければならず、ベクトル数もその
分多くなってしまふ。

【0007】さらに、たとえば、図17に示すように、枠線を持つ閉領域を従来の方法でベクトル変換した後に1.5倍に拡大すると、上記のように2つの閉領域としてベクトル化されてしまうため、枠線の線幅が変わってしまう。これは、縮小の場合も同様である。

【0008】また、上記のように、線分として描かれる図形が、ベクトル変換後に4本のベクトルで囲まれた閉領域として出力され（図16参照）、または、枠線を持つ閉領域が、ベクトル変換後に2つの閉領域として出力されるため（図17参照）、アプリケーションソフトでデータを再利用する際に加工（編集）しにくい。

【0009】要するに、従来のベクトル変換技術にあっては、データ量の低減には一定の限界があるとともに、拡大／縮小の編集作業において線幅が変わってしまう場合があり、さらに、データの再利用時に加工（編集）しにくいという問題がある。

【0010】本発明は、このような従来技術の問題点を鑑みてなされたものであり、図形の色を再現することができるとともに、ベクトル変換後のデータ量をより一層低減することができ、データの編集時に枠線付き閉領域を拡大／縮小しても線幅が変わることがなく、データの再利用時における使い勝手の向上をも図ることができる画像処理装置および方法ならびに画像処理プログラムを記録したコンピュータ読取可能な記録媒体を提供することを目的とする。

【0011】

【課題を解決するための手段】本発明の上記目的は、下記的手段によって達成される。

【0012】（1）本発明に係る画像処理装置は、図形をベクトルデータに変換するベクトル変換手段と、ベクトルデータに変換された線図形の線幅を検出する線幅検出手段と、変換されたベクトルデータに基づいて閉領域を抽出する閉領域抽出手段と、検出された線幅の情報に応じて、抽出された閉領域の内部の色を検出する閉領域色検出手段とを有することを特徴とする。

【0013】（2）検出された線幅の情報は、対応する線図形のベクトルデータに付加される。

【0014】（3）検出された線幅が設定値以上であるか否かを判断する判断手段をさらに有し、前記ベクトル変換手段は、検出された線幅が設定値以上である場合、対応する線図形のエッジを検出し、検出したエッジをベクトルデータに変換する。

【0015】（4）前記閉領域色検出手段は、検出された線幅の情報に応じて、抽出された閉領域の内部点であって当該閉領域を形成するベクトルデータに対応する線図形よりも内側に存在する内部点を選択する選択手段を含み、選択された内部点の色情報に基づいて、抽出された閉領域の内部の色を検出する。

【0016】（5）ベクトルデータに変換された線図形の色を検出する線色検出手段をさらに有する。

【0017】（6）検出された色の情報は、対応する線図形のベクトルデータに付加される。

【0018】（7）上記の画像処理装置は、画像読取装置に搭載されている。

【0019】（8）本発明に係る画像処理方法は、図形をベクトルデータに変換するステップと、ベクトルデータに変換された線図形の線幅を検出するステップと、変換されたベクトルデータに基づいて閉領域を抽出するステップと、検出された線幅の情報に応じて、抽出された閉領域の内部の色を検出するステップとを有することを特徴とする。

【0020】（9）本発明に係るコンピュータ読取可能な記録媒体は、画像処理プログラムを記録したコンピュータ読取可能な記録媒体であって、該画像処理プログラムが、図形をベクトルデータに変換するステップと、ベクトルデータに変換された線図形の線幅を検出するステップと、変換されたベクトルデータに基づいて閉領域を抽出するステップと、検出された線幅の情報に応じて、抽出された閉領域の内部の色を検出するステップとを有することを特徴とする。

【0021】

【発明の実施の形態】以下、図面を使って本発明の実施の形態を説明する。

【0022】（第1の実施の形態）図1は、本発明の第1の実施の形態に対応するデジタル複写機の構成を示すブロック図である。すなわち、第1の実施の形態では、本発明に係る画像処理装置は、デジタル複写機として構成されている。

【0023】このデジタル複写機100は、スキャナとプリンタの複合機であって、カラー原稿中の図形（塗り潰された閉領域を含む）を自動的に少ないデータ量で精度の良いベクトルデータに変換することができるベクトル変換機能を有する。なお、本発明によるベクトル変換機能は、後述するように、単に線図形をベクトル化するだけでなく、線幅を検出し、閉領域を抽出し、そしてベクトルや閉領域の色を検出して、これらの情報をベクトルデータに付加することができるという高度の機能を有するので、ここでは、従来のベクトル変換機能と区別するために、「高度ベクトル変換機能」と呼ぶこととする。

【0024】デジタル複写機100は、各種の入力と表示を行う操作パネル1と、原稿を読み取ってデジタル画像データ（ラスタデータ）に変換するスキャナエンジン3と、複数のシート状原稿を自動的に1枚ずつ原稿セット位置まで搬送するADF（自動原稿搬送装置）5と、読み取って得られた画像データを処理する画像処理部7と、PDFなどのファイル形式で記録された画像処理後のデータ（後述するようにベクトルデータを含む）を画面出力／印刷出力用のビットマップデータ（ラスタデータ）に展開するラスタライザ9と、ビットマップ方式の

画像データを用紙に印刷するプリンタエンジン 11 と、プログラムやデータを記憶するメモリ 13 と、画像処理後のファイルをネットワーク（図示せず）を介して外部へ出力するための通信インタフェース 15 と、上記の各部を制御する制御部 17 とを有する。

【0025】操作パネル 1 は、図示しないが、タッチパネルディスプレイやスタートキー、10 キー（テンキー）などを有する。タッチパネルディスプレイの画面には、現在設定されているモード、選択できるモード、当該複写機 100 の状態などがグラフィックスやメッセージで表示される。ユーザは、画面内に表示されたキーをタッチすることで、表示画面やコピーモードを切り替えることができる。

【0026】スキャナエンジン 3 は、カラー原稿を読み取ることができるカラー対応のスキャナエンジンである。ラスタライザ 9 は、たとえば、RIP（リップ）で構成されている。プリンタエンジン 11 は、カラー印刷が可能なプリンタエンジンであって、たとえば、カラー対応のレーザプリンタ（LBP）エンジンやインクジェットプリンタエンジンなどで構成されている。

【0027】メモリ 13 は、図示しないが、プログラムやデータを記憶する ROM と、データを一時的に記憶する RAM とで構成されている。ROM は、プログラムやデータのアップデートが可能なように、主として、電気的に内容を書き直せるタイプのフラッシュ ROM（フラッシュメモリとも呼ばれる）で構成されている。画像処理を行うための各種パラメータは、ROM に記憶されている。各部で処理されたデータは、RAM に一時的に記憶される。

【0028】通信インタフェースは、たとえば、ネットワークインタフェースカード（NIC：ニック）（LAN ボードとも呼ばれる）で構成されている。また、制御部 17 は、CPU で構成されている。

【0029】画像処理部 7 は、スキャナエンジン 3 で読み取って得られた原稿画像の文字領域、図形領域、絵柄領域（写真領域）を判別し、分割する領域判別部 19 と、文字領域に含まれる文字画像を認識して、文字データ（たとえば、JIS コードなどの文字コード）に変換する文字認識部 21 と、図形領域に含まれる線図形（線分や曲線）を認識して、線幅の情報を持つベクトルデータに変換するベクトル変換部 23 と、ベクトルデータに基づいて閉領域を抽出する閉領域抽出部 27 と、ベクトルデータに付加される色情報（ベクトルの色、閉領域の色）を検出する色検出部 29 と、絵柄領域に対してスムージングやエッジ強調などの前処理および誤差拡散法（ディザ法）などの二値化処理を含むコピー画像処理（原画像を忠実に再現するための画像処理という意味）を行うコピー画像処理部 35 と、文字データ、ベクトルデータ（線幅と色の情報を持つ）および絵柄領域を合成し、合成後の画像データを所定のファイル形式（たと

ば、PDF など）で記録した画像ファイルを作成する合成処理部 37 とを有する。さらに、ベクトル変換部 23 は、線図形の線幅を検出する線幅検出部 25 を含み、色検出部 29 は、ベクトルの色を検出するベクトル色検出部 31 と、閉領域の色を検出する閉領域色検出部 33 とで構成されている。

【0030】図 2 は、このようなデジタル複写機 100 の動作を示すメインフローチャートである。図 2 に示すフローチャートは、デジタル複写機 100 のメモリ 13（ROM）に制御プログラムとして記憶されており、制御部 17（CPU）によって実行される。

【0031】まず、ステップ S1000 では、ユーザによって、操作パネル 1 上の 10 キーやタッチパネルディスプレイに表示された設定画面を使ってコピーモードが設定される。コピーモードには、たとえば、コピー枚数、コピー用紙、コピー倍率、コピー濃度、片面／両面コピー、ノンソート／ソートなどがある。この時、ユーザは、併せて、ADF 5 を使用してまたは使用せずに、コピーをとる原稿を所定の方法でセットしておく。

【0032】そして、ステップ S2000 では、操作パネル 1 上のスタートキーが押されたか否かを判断する。スタートキーが押されていない場合は（S2000：NO）、スタートキーが押されるまで待機し、スタートキーが押された場合は（S2000：YES）、ユーザの指示があったものと判断して、ステップ S3000 に進み、コピー動作を開始する。

【0033】ステップ S3000 では、スキャナエンジン 3 で、セットされた原稿を読み取ってデジタル画像データ（ラスタデータ）に変換する。ADF 5 を使用する場合は、複数枚の原稿が自動的に 1 枚ずつ送り出され、連続的に読取り動作が行われる。得られた画像データは、色の情報を持っている。

【0034】そして、ステップ S4000 では、画像処理部 7 で、ステップ S3000 で得られた画像データに対して所定の画像処理（後で詳述する）を行い、出力用の画像ファイルを作成する。

【0035】そして、ステップ S5000 では、ステップ S4000 で作成された画像ファイルをラスタライザ 9 でビットマップデータ（ラスタデータ）に展開した後、プリンタエンジン 11 で用紙に印刷する。このとき、セットされた原稿がカラー原稿の場合は、カラー印刷が行われる。

【0036】図 3 は、図 2 の画像処理の内容を示すフローチャートである。なお、図 3 に示すフローチャートは、図 2 に示すフローチャートのサブルーチンとして、メモリ 13（ROM）に記憶されている。

【0037】まず、ステップ S4100 では、領域判別部 19 で、ステップ S3000 で得られた画像データに対して領域判別処理を行う。具体的には、スキャナエンジン 3 で読み取って得られた原稿画像（以下「入力画

10

20

30

40

50

像」という)を、文字領域、図形領域、絵柄領域(写真領域)の3種類の領域に分類し、分割する。そして、以下に示すように、領域ごとに別々の処理を行う。

【0038】すなわち、まず、ステップS4200で、文字領域か否かを判断する。文字領域の場合は(S4200:YES)、ステップS4300に進み、文字領域でない場合は(S4200:NO)、ステップS4400に進む。

【0039】文字領域の場合

ステップS4300では、文字認識部21で、文字領域に対して文字認識処理を行う。具体的には、文字領域に含まれる文字画像を認識して、文字データ(たとえば、JISコードなどの文字コード)に変換する。

【0040】一方、ステップS4400では、図形領域か否かを判断する。図形領域の場合は(S4400:YES)、ステップS4500に進み、図形領域でない、つまり、絵柄領域の場合は(S4400:NO)、ステップS4800に進む。

【0041】図形領域の場合

ステップS4500では、ベクトル変換部23で、図形領域に対してベクトル変換処理を行う。すなわち、図形領域に含まれる線図形(線分や曲線)を認識して、線幅の情報を持つベクトルデータに変換する。ここで、線幅の情報は、線幅検出部25によって検出される。また、ベクトル自体は、始点と終点の座標で表現されている。この処理の具体的な内容は、後で詳述する。

【0042】そして、ステップS4600では、閉領域抽出部27で、ベクトル化されたデータに対して閉領域抽出処理を行う。すなわち、ステップS4500で変換されたベクトルデータに基づいて、閉領域を抽出する。この処理の具体的な内容は、後で詳述する。

【0043】そして、ステップS4700では、色検出部29で、色検出処理を行う。すなわち、ステップS4500で変換されたベクトルデータに付加される色情報(ベクトルの色、閉領域の色)を検出する。ベクトルの色は、ベクトル色検出部31によって、閉領域の色は、閉領域色検出部33によってそれぞれ検出される。この処理の具体的な内容は、後で詳述する。

【0044】なお、図形領域に対する上記一連の処理(ステップS4500のベクトル変換処理、ステップS4600の閉領域抽出処理、ステップS4700の色検出処理)は、本発明に係る高度ベクトル変換機能を実現するための処理であり、ここでは、これら3つの処理をまとめて「高度ベクトル変換処理」と呼ぶこととする。よって、高度ベクトル変換処理の結果得られるベクトルデータは、ベクトル(始点と終点の座標)の情報に加えて、線幅および色の情報を持つことになる。

【0045】絵柄領域の場合

ステップS4800では、コピー画像処理部35で、絵柄領域に対してコピー画像処理、具体的には、スムージ

ングやエッジ強調などの前処理および誤差拡散法などの二値化処理を含んだ、原画像を忠実に再現するための画像処理を行う。

【0046】そして、すべての領域に対する処理が終わると、ステップS4900では、合成処理部37で、各部の処理結果である文字データ、ベクトルデータ(線幅と色の情報を持つ)および絵柄領域を合成し、合成後の画像データを所定のファイル形式(PDFなど)で記録した画像ファイルを作成する。この画像ファイルは、メモリ13(RAM)に記憶される。

【0047】図4は、図3のベクトル変換処理の内容を示すフローチャートである。なお、図4に示すフローチャートは、図3に示すフローチャートのサブルーチンとして、メモリ13(ROM)に記憶されている。

【0048】まず、ステップS4505では、図形領域の画像データに対して任意の適当な方法で二値化処理を行い、二値画像を得る。

【0049】そして、ステップS4510では、ステップS4505で得られた二値画像に対して任意の適当な方法で芯線化処理を行い、線幅1の中心線(芯線)を抽出する。

【0050】そして、ステップS4515では、ステップS4510で得られた芯線データを用いてベクトル化を行う。すなわち、芯線をベクトル(始点と終点の座標で表現される)に変換する(「芯線ベクトル」という)。このとき、芯線は、一般的には線分のベクトルの集まりで表現される。直線部分が長いとベクトルが長くなり、曲線の部分は短いベクトルの連続する集まりで表される。

【0051】なお、ここでは、線図形をベクトル化する方法として、芯線を抽出してベクトルにする方法を用いているが、ベクトル化の方法はこれに限定されるわけではなく、任意の適当な方法を用いることができる。

【0052】そして、ステップS4520では、ステップS4515でベクトル化された線図形の線幅を検出する。線幅の検出は、具体的には、たとえば、芯線化またはベクトル化の処理の中で得られるデータ(たとえば、線図形のエッジ部と芯線との距離に関するデータ)を統計処理する(たとえば、平均値を求めて2倍する)ことによって行われる。検出された線幅の情報は、対応する線図形のベクトルデータに付加される。なお、線幅の検出方法についても、任意の適当な方法を用いることができる。

【0053】そして、ステップS4525では、ステップS4515で得られたすべてのベクトル(芯線ベクトル)の中から1つのベクトルを選択し、選択されたベクトルに付加されているステップS4520で検出された線幅をあらかじめ設定されたしきい値と比較して、その線幅がしきい値よりも大きいかな否かを判断する。これは、線図形をそのまま線として扱ってもよいかそれとも

領域として扱うべきかを判別するためである。線幅がしきい値よりも大きい場合は (S 4 5 2 5 : YES)、線ではなく領域として扱うべきと判断して、ステップ S 4 5 3 0 に進み、線幅がしきい値以下である場合は (S 4 5 2 5 : NO)、そのまま線として扱ってもよいと判断して、ただちにステップ S 4 5 4 0 に進む。

【0054】ステップ S 4 5 3 0 では、その線幅を持つベクトルに対応する線図形の入力画像に対して任意の適当な方法でエッジの検出を行う。

【0055】そして、ステップ S 4 5 3 5 では、ステップ S 4 5 3 0 で検出されたエッジデータを用いてベクトル化を行う。すなわち、検出されたすべてのエッジをベクトルに変換する(「エッジベクトル」という)。エッジベクトルには、線幅が 1 である旨の情報が付与される。

【0056】そして、ステップ S 4 5 4 0 では、ステップ S 4 5 1 5 で得られたすべてのベクトル(芯線ベクトル)に対してステップ S 4 5 2 5 の比較処理が終了したか否かを判断する。すべての芯線ベクトルに対して比較処理が終了していない場合は (S 4 5 4 0 : NO) は、ステップ S 4 5 2 5 に戻って、次の芯線ベクトルに対する比較処理を開始し、すべての芯線ベクトルに対して比較処理が終了している場合は (S 4 5 4 0 : YES)、図 3 に示すフローチャートにリターンする。

【0057】図 5 は、図 3 の閉領域抽出処理の内容を示すフローチャートである。なお、図 5 に示すフローチャートは、図 3 に示すフローチャートのサブルーチンとして、メモリ 1 3 (ROM) に記憶されている。

【0058】まず、ステップ S 4 6 0 5 では、注目ベクトルの始点の座標をメモリ 1 3 (RAM) に保存する。なお、当該閉領域抽出処理の開始時には、ステップ S 4 5 0 0 で得られたベクトル化されたデータ(ベクトル群)の中から 1 つのベクトルを選択し、これを注目ベクトルとする。また、ここでは、始点座標が保存される初期段階の注目ベクトルのことを「初期注目ベクトル」と呼んで、始点座標の保存を伴わない単なる注目ベクトルと区別することとする。

【0059】そして、ステップ S 4 6 1 0 では、注目ベクトルの終点に連結する他のベクトルが存在するか否かを判断する。この判断は、注目ベクトルの終点座標と同一の始点座標または終点座標を有するベクトルを、ベクトル群の中から探索することによって行われる。注目ベクトルの終点に連結する他のベクトルが存在する場合は (S 4 6 1 0 : YES)、ステップ S 4 6 1 5 に進み、注目ベクトルの終点に連結する他のベクトルが存在しない場合は (S 4 6 1 0 : NO)、ただちにステップ S 4 6 5 5 に進む。

【0060】ステップ S 4 6 1 5 では、ステップ S 4 6 1 0 で探索された他のベクトルの中から 1 つのベクトルを選択し、これを候補ベクトルとする。

【0061】そして、ステップ S 4 6 2 0 では、候補ベクトル以外に注目ベクトルの終点に連結するベクトルが存在するか否かを判断する。候補ベクトル以外に連結ベクトルが存在する場合は (S 4 6 2 0 : YES)、ステップ S 4 6 2 5 に進み、候補ベクトル以外に連結ベクトルが存在しない場合は (S 4 6 2 0 : NO)、ただちにステップ S 4 6 3 0 に進む。

【0062】ステップ S 4 6 2 5 では、初期注目ベクトルから現在の注目ベクトルまでのルートを示すベクトル群(「注目ベクトルまでのベクトル群」という)と候補ベクトル以外の連結ベクトル群とをメモリ 1 3 (RAM) に設けられたスタックに保存する。これは、ベクトルの分岐によって閉領域の抽出ができなくなることを防止するためである。

【0063】そして、ステップ S 4 6 3 0 では、候補ベクトルが今まで一度も使用されていないか否かを判断する。候補ベクトルが今まで一度も使用されていない場合は (S 4 6 3 0 : YES)、ステップ S 4 6 3 5 に進み、候補ベクトルが今までに使用されたことがある場合は (S 4 6 3 0 : NO)、ただちにステップ S 4 6 5 5 に進む。

【0064】ステップ S 4 6 3 5 では、候補ベクトルの終点の座標とステップ S 4 6 0 5 で保存された始点の座標とが一致するか否かを判断する。候補ベクトルの終点座標と保存されている始点座標とが一致しない場合は (S 4 6 3 5 : NO)、ステップ S 4 6 4 0 に進み、候補ベクトルの終点座標と保存されている始点座標とが一致する場合は (S 4 6 3 5 : YES)、閉領域が検出されたものと判断して、ステップ S 4 6 4 5 に進む。

【0065】ステップ S 4 6 4 0 では、候補ベクトルを注目ベクトルとして、ステップ S 4 6 1 0 に戻り、以上の処理を繰り返し行う。

【0066】一方、ステップ S 4 6 4 5 では、ステップ S 4 6 3 5 で検出された閉領域が今までに抽出された既にグループ化されている閉領域と同一か否かを判断する。この判断は、閉領域を構成するベクトル群同士を比較することによって行われる。検出された閉領域が既にグループ化されている閉領域と同一である場合は (S 4 6 4 5 : YES)、ただちにステップ S 4 6 5 5 に進み、検出された閉領域が既にグループ化されている閉領域と同一でない場合は (S 4 6 4 5 : NO)、ステップ S 4 6 5 0 に進む。

【0067】ステップ S 4 6 5 0 では、検出されたベクトル群を新しい閉領域としてグループ化する。

【0068】そして、ステップ S 4 6 5 5 では、スタックに何かデータが保存されているか否かを判断する。この判断は、上記のように、注目ベクトルの終点に連結する他のベクトルが存在しない場合 (S 4 6 1 0 : NO)、候補ベクトルが今までに使用されたことがある場合 (S 4 6 3 0 : NO)、既にグループ化されている閉

領域と同一の閉領域を検出した場合（S4645：YES）にそれぞれ行われる。スタックにデータが保存されている場合は（S4655：YES）、ステップS4660に進み、スタックにデータが保存されていない場合は（S4655：NO）、スタックに保存されているデータをすべて探索し終えたものと判断して、ステップS4670に進む。

【0069】ステップS4660では、スタックからデータを読み出す。

【0070】そして、ステップS4665では、読み出されたスタック内の連結ベクトル群の中から1つのベクトルを選択し、これを候補ベクトルとして、ステップS4620に戻る。これにより、分岐による探索漏れをなくすることができる。

【0071】一方、ステップS4670では、ステップS4500で得られたベクトル群の中のすべてのベクトルに対して閉領域の抽出が終了したか否かを判断する。全ベクトルに対して閉領域の抽出が終了していない場合は（S4670：NO）、ステップS4675に進み、全ベクトルに対して閉領域の抽出が終了した場合は（S4670：YES）、当該閉領域抽出処理を終了して、図3のフローチャートにリターンする。

【0072】ステップS4675では、ステップS4500で得られたベクトル群の中から次の1つのベクトルを選択し、これを初期注目ベクトルとする。そして、ステップS4605に戻り、以上の処理を繰り返し行う。

【0073】次に、図9～図14を参照して、閉領域抽出処理の内容をさらに具体的に説明する。ここで、図9は、ベクトル化されたデータ（ベクトル群）を示す図であり、図10～図14は、スタックに保存されているデータを示す図である。

【0074】まず、ベクトルABを初期注目ベクトルとして選択し、この初期注目ベクトルABの始点Aの座標をメモリ13（RAM）に保存する（S4605）。そして、この注目ベクトルABの終点Bに連結する他のベクトルを探索し、この場合、注目ベクトルABの終点Bに連結する他のベクトルが2つ存在するため（S4610：YES）、その中の1つのベクトルBCを候補ベクトルとして選択し（S4615）、さらに、候補ベクトルBC以外の連結ベクトル（ベクトルBE）が存在するので（S4620：YES）、注目ベクトルまでのベクトル群（ベクトルAB）と候補ベクトル以外の連結ベクトル群（ベクトルBE）とをスタックに保存する（図10参照）（S4625）。そして、候補ベクトルBCは一度も使用されておらず（S4630：YES）、候補ベクトルBCの終点Cは保存されている始点Aと一致していないので（S4635：NO）、候補ベクトルBCを注目ベクトルとする（S4640）。

【0075】そして、注目ベクトルBCの終点Cに連結する他のベクトルを探索し、この場合、注目ベクトルB

Cの終点Cに連結する他のベクトルが1つ存在するため（S4610：YES）、このベクトルCDを候補ベクトルとして選択し（S4615）、この候補ベクトルCD以外には連結ベクトルは存在しないので（S4620：NO）、スタックへのデータの保存は行わない。そして、候補ベクトルCDは一度も使用されておらず（S4630：YES）、候補ベクトルCDの終点Dは保存されている始点Aと一致していないので（S4635：NO）、候補ベクトルCDを注目ベクトルとする（S4640）。

【0076】そして、注目ベクトルCDの終点Dに連結する他のベクトルを探索し、この場合、注目ベクトルCDの終点Dに連結する他のベクトルは存在せず（S4610：NO）、スタックにはデータが保存されているので（図10参照）（S4655：YES）、スタックからデータを読み出し（S4660）、連結ベクトル群の中から1つのベクトルBEを選択し、候補ベクトルとする（S4665）。この時点で、一旦、スタックは空になる。

【0077】この場合、候補ベクトルBE以外には連結ベクトルは存在しないので（S4620：NO）、スタックへのデータの保存は行わない。そして、候補ベクトルBEは一度も使用されておらず（S4630：YES）、候補ベクトルBEの終点Eは保存されている始点Aと一致していないので（S4635：NO）、候補ベクトルBEを注目ベクトルとする（S4340）。

【0078】そして、注目ベクトルBEの終点Eに連結する他のベクトルを探索し、この場合、注目ベクトルBEの終点Eに連結する他のベクトルが3つ存在するため（S4610：YES）、その中の1つのベクトルEFを候補ベクトルとして選択し（S4615）、さらに、候補ベクトルEF以外にも連結ベクトル（ベクトルEG、ベクトルEH）が存在するので（S4620：YES）、注目ベクトルまでのベクトル群（ベクトルAB、ベクトルBE）と候補ベクトル以外の連結ベクトル群（ベクトルEG、ベクトルEH）とをスタックに保存する（図11参照）（S4625）。そして、候補ベクトルEFは一度も使用されておらず（S4630：YES）、候補ベクトルEFの終点Fは保存されている始点Aと一致していないので（S4635：NO）、候補ベクトルEFを注目ベクトルとする（S4640）。

【0079】そして、注目ベクトルEFの終点Fに連結する他のベクトルを探索し、この場合、注目ベクトルEFの終点Fに連結する他のベクトルが1つ存在するため（S6105：YES）、このベクトルFGを候補ベクトルとして選択し（S4615）、この候補ベクトルFG以外には連結ベクトルは存在しないので（S4620：NO）、スタックへのデータの保存は行わない。そして、候補ベクトルFGは一度も使用されておらず（S4630：YES）、候補ベクトルFGの終点Gは保存

されている始点Aと一致していないので（S4635：NO）、候補ベクトルFGを注目ベクトルとする（S4640）。

【0080】そして、注目ベクトルFGの終点Gに連結する他のベクトルを探索し、この場合、注目ベクトルFGの終点Gに連結する他のベクトルが1つ存在するため（S4610：YES）、このベクトルGEを候補ベクトルとして選択し（S4615）、この候補ベクトルGE以外には連結ベクトルは存在しないので（S4620：NO）、スタックへのデータの保存は行わない。そして、候補ベクトルGEは一度も使用されておらず（S4630：YES）、候補ベクトルGEの終点Eは保存されている始点Aと一致していないので（S4635：NO）、候補ベクトルGEを注目ベクトルとする（S4640）。

【0081】そして、注目ベクトルGEの終点Eに連結する他のベクトルを探索し、この場合、注目ベクトルGEの終点Eに連結する他のベクトルが3つ存在するため（S4610：YES）、その中の1つのベクトルEBを候補ベクトルとして選択し（S4615）、さらに、候補ベクトルEB以外にも連結ベクトル（ベクトルEF、ベクトルEH）が存在するので（S4620：YES）、注目ベクトルまでのベクトル群（ベクトルAB、ベクトルBE、ベクトルEF、ベクトルFG、ベクトルGE）と候補ベクトル以外の連結ベクトル群（ベクトルEF、ベクトルEH）とをスタックに保存する（図12参照）（S4625）。しかし、候補ベクトルEBはこれまでのルートで使用されており（S4630：NO）、スタックにはデータが保存されているので（図12参照）（S4655：YES）、スタックからデータを読み出し（S4660）、連結ベクトル群の中から1つのベクトルEFを選択し、候補ベクトルとする（S4665）。この時点で、スタックに保存されているデータは、一旦、図11に示すようになる。

【0082】この場合、候補ベクトルEF以外の他の連結ベクトル（ベクトルEH）が存在するので（S4620：YES）、注目ベクトルまでのベクトル群（ベクトルAB、ベクトルBE、ベクトルEF、ベクトルFG、ベクトルGE）と候補ベクトル以外の連結ベクトル群（ベクトルEH）とをスタックに保存する（図13参照）（S4625）。しかし、候補ベクトルEFもこれまでのルートで使用されており（S4630：NO）、スタックにはデータが保存されているので（図13参照）（S4655：YES）、スタックからデータを読み出し（S4660）、連結ベクトル群の中から1つのベクトルEHを選択し、候補ベクトルとする（S4665）。この時点で、スタックに保存されているデータは、再度、図11に示すようになる。

【0083】この場合、候補ベクトルEH以外には連結ベクトルは存在しないので（S4620：NO）、スタ

ックへのデータの保存は行わない。そして、候補ベクトルEHは一度も使用されておらず（S4630：YES）、候補ベクトルEHの終点Hは保存されている始点Aと一致していないので（S4635：NO）、候補ベクトルEHを注目ベクトルとする（S4640）。

【0084】そして、注目ベクトルEHの終点Hに連結する他のベクトルを探索し、この場合、注目ベクトルEHの終点Hに連結する他のベクトルは存在せず（S4610：NO）、スタックにはデータが保存されているので（図11参照）（S4655：YES）、スタックからデータを読み出し（S4660）、連結ベクトル群の中から1つのベクトルEGを選択し、候補ベクトルとする（S4665）。この時点で、再度、スタックは空になる。

【0085】この場合、候補ベクトルEG以外の他の連結ベクトル（ベクトルEH）が存在するので（S4620：YES）、注目ベクトルまでのベクトル群（ベクトルAB、ベクトルBE）と候補ベクトル以外の連結ベクトル群（ベクトルEH）とをスタックに保存する（図14参照）（S4625）。しかし、候補ベクトルEGはこれまでのルートで使用されており（S4630：NO）、スタックにはデータが保存されているので（図14参照）（S4655：YES）、スタックからデータを読み出し（S4660）、連結ベクトル群の中から1つのベクトルEHを選択し、候補ベクトルとする（S4665）。この時点で、スタックは空になる。

【0086】この場合、候補ベクトルEH以外には連結ベクトルは存在しないので（S4620：NO）、スタックへのデータの保存は行わない。そして、候補ベクトルEHはこれまでのルートで使用されており（S4630：NO）、スタックにデータは何も保存されていないので（S4655：NO）、ベクトルABに対する閉領域の抽出が終了する。しかし、全ベクトルに対して閉領域の抽出は終了していないので（S4670：NO）、ベクトル群（図9参照）の中から次の1つのベクトル（たとえば、ベクトルBC）を初期注目ベクトルとして選択し（S4675）、ステップS4605に戻る。

【0087】そして、残りの各ベクトルに対して閉領域の抽出を順次行くと、初期注目ベクトルが、ベクトルEF、ベクトルFG、ベクトルEGの時に閉領域EFGが検出される。ただし、一番最初に検出された時だけ（S4645：NO）、閉領域としてグループ化される（S4650）。

【0088】図6は、図3の色検出処理の内容を示すフローチャートである。なお、図6に示すフローチャートは、図2に示すフローチャートのサブルーチンとして、メモリ13（ROM）に記憶されている。

【0089】まず、ステップS4710では、ステップS4500で得られた各ベクトルの色（つまり、ベクトル化された線の色）を検出する。

【0090】そして、ステップS4750では、ステップS4600で抽出された閉領域の色（つまり、ベクトル化された線（枠線）で囲まれた閉領域の色）を検出する。

【0091】図7は、図6のベクトル色検出処理の内容を示すフローチャートである。なお、図7に示すフローチャートは、図6に示すフローチャートのサブルーチンとして、メモリ13（ROM）に記憶されている。

【0092】このベクトル色検出処理を実行するに当たっては、ステップS4500で得られたすべてのベクトルデータがメモリ13（RAM）から読み出される。なお、ベクトルデータは、前述のように、始点と終点の座標データからなり、この段階では、線幅の情報も付加されている。

【0093】まず、ステップS4715では、1つのベクトル（注目ベクトル）について、その始点座標と終点座標からベクトル上の中点（始点と終点を1:1に内分する点）の座標を計算する。

【0094】そして、ステップS4720では、入力画像データから、始点、終点、中点の3点の座標に対応する位置の色情報（RGB、CMYK、L*a*b*など）をそれぞれ取得する。

【0095】そして、ステップS4725では、始点、終点、中点の色情報の平均値を計算し、この平均値を当該ベクトルの色とする。

【0096】そして、ステップS4730では、すべてのベクトルに対して色の検出が終了したか否かを判断する。全ベクトルに対して色の検出が終了していない場合は（S4730:NO）、ステップS4735に進み、全ベクトルに対して色の検出が終了した場合は（S4730:YES）、当該ベクトル色検出処理を終了して、図6のフローチャートにリターンする。

【0097】ステップS4735では、次のベクトルを注目ベクトルとして選択して、ステップS4715に戻る。

【0098】図8は、図6の閉領域色検出処理の内容を示すフローチャートである。なお、図8に示すフローチャートは、図6に示すフローチャートのサブルーチンとしてメモリ13（ROM）に記憶されている。

【0099】この閉領域色検出処理を実行するに当たっては、ステップS4600で得られた、閉領域としてグループ化されたすべてのベクトルデータ群が、メモリ13（RAM）から読み出される。なお、ベクトルデータは、前述のように、始点と終点の座標データに加えて、線幅の情報を有している。

【0100】まず、ステップS4755では、ある閉領域（注目閉領域）を構成するベクトル群の中の1つのベクトル（注目ベクトル）に対して、ベクトル上の1点からベクトルに垂直な方向に閉領域の内部へ所定の距離、たとえば、 $\{(線幅/2) + \alpha\}$ ドットだけ進んだ点

（内部点）の座標を計算する。なお、内部点の座標の計算式は、もちろん、 $\{(線幅/2) + \alpha\}$ の式に限定されるわけではなく、線幅を含まない閉領域の内部点の座標を選択できる式であればどのような式でもよい。

【0101】そして、ステップS4760では、入力画像データから、ステップS4755で得られた内部点の座標に対応する位置の色情報（RGB、CMYK、L*a*b*など）を取得する。

【0102】そして、ステップS4765では、現在の注目ベクトルの長さ分、具体的には、その注目ベクトルに対する所定の数のサンプリング点について、色情報の取得が終了したか否かを判断する。注目ベクトルの長さ分について色情報の取得が終了した場合は（S4765:YES）、ステップS4770に進み、注目ベクトルの長さ分について色情報の取得が終了していない場合は（S4765:NO）、ステップS4755に戻る。

【0103】ステップS4770では、現在の注目閉領域を構成するベクトル群の中のすべてのベクトルに対して、ベクトル長さ分の色情報の取得が終了したか否かを判断する。閉領域を構成する全ベクトルに対してベクトル長さ分の色情報の取得が終了していない場合は（S4770:NO）、ステップS4775に進み、閉領域を構成する全ベクトルに対してベクトル長さ分の色情報の取得が終了した場合は（S4770:YES）、ステップS4780に進む。

【0104】ステップS4775では、現在の注目閉領域を構成するベクトル群の中の次のベクトルを注目ベクトルとして選択して、ステップS4755に戻る。

【0105】ステップS4780では、閉領域を構成する全ベクトルについてそれぞれ取得されたベクトル長さ分の各色情報に対して、ヒストグラムを作成する。

【0106】そして、ステップS4785では、ステップS4780で作成されたヒストグラムにおいて頻度が最も高い色情報の値（最多値）を求め、この最多値を当該閉領域の色とする。ここで、閉領域の色として最多値を用いるのは、次の理由による。ベクトルの交点付近は、閉領域内部の色ではなく、垂直に交わる線の色自体をピックアップする可能性があるため、ベクトル交点付近のデータを無視するために、ここでは最多値を閉領域の色としている。

【0107】そして、ステップS4790では、抽出されたすべての閉領域に対して色の検出が終了したか否かを判断する。すべての閉領域に対して色の検出が終了していない場合は（S4790:NO）、ステップS4795に進み、すべての閉領域に対して色の検出が終了した場合は（S4790:YES）、当該閉領域色検出処理を終了して、図6のフローチャートにリターンする。

【0108】ステップS4795では、抽出された閉領域の中の次の閉領域を注目閉領域として選択して、ステップS4755に戻る。

【0109】なお、本実施の形態では、色の検出に当たって平均値（ベクトル色検出処理）や最多値（閉領域検出処理）を用いているが、これに限定されるわけではなく、たとえば、中央値（メディアン）などの他の統計値を用いてもよい。また、いずれの処理においても、色情報を取得するためのサンプリング点を多くとればとるほど、より正確な値を得ることができる。

【0110】次に、図15を用いて、図形領域に対する上記処理の具体例を説明する。ここでは、図15（A）に示すように、図形領域として黒色の枠線を持つ閉領域で内部が緑色で塗り潰されたものを例にとって説明する。なお、枠線の線幅は、しきい値よりも小さいとする。

【0111】まず、図15（A）に示す枠線付き閉領域の画像データがベクトル変換部23に入力されると、図3のステップS4500（図4参照）で、順次、二値化（S4505）、芯線化（S4510）、ベクトル化（S4515）、線幅検出（S4520）が行われる。そして、この場合、線幅がしきい値よりも小さい（S4525：NO）ので、芯線ベクトルが出力される。具体的には、芯線ベクトルとして、図15（B）に示すように、ベクトルEF、ベクトルFG、ベクトルGH、ベクトルHEの4本のベクトルが出力される。これらのベクトルデータは、それぞれ、始点の座標、終点の座標、および線幅の情報を有している。

【0112】そして、この4本分のベクトルデータが閉領域抽出部27に入力されると、図3のステップS4600（図5参照）で、まず、たとえば、ベクトルEFを注目ベクトル（初期注目ベクトル）として設定する。そして、ベクトルEFの始点Eの座標をメモリ13（RAM）に保存する（S4605）。そして、この場合、終点Fに連結する他のベクトルが1つ存在するので（S4610：YES）、このベクトルFGを候補ベクトルとする（S4615）。このベクトルFGはまだ一度も使用されておらず（S4630：YES）、ベクトルFGの終点Gの座標と保存されている始点Eの座標とは一致していないので（S4635：NO）、ベクトルFGを注目ベクトルとする（S4640）。

【0113】そして、注目ベクトルFGについて同様の処理を行うと、今度は注目ベクトルがベクトルGFとなり、さらに、処理を続けると、今度は注目ベクトルがベクトルHEとなる。注目ベクトルがベクトルHEの場合、ベクトルHEの終点Eと保存されている始点Eの座標とが一致するので（S4635：YES）、このベクトル群（ベクトルEF、ベクトルFG、ベクトルGH、ベクトルHE）が既に閉領域としてグループ化されているか否かを判断する（S4645）。この場合、まだグループ化されていないので（S4645：NO）、このベクトル群（ベクトルEF、ベクトルFG、ベクトルGH、ベクトルHE）は、閉領域としてグループ化される

（S4650）。

【0114】そして、色検出部29（ベクトル色検出部31、閉領域色検出部33）にて、図3のステップS4700（図6～図7参照）で、ベクトル自体の色と閉領域内部の色をそれぞれ検出する。

【0115】ベクトル色検出処理では、4本分のベクトルデータがベクトル色検出部31に入力されると、図6のステップS4710（図7参照）で、まず、たとえば、ベクトルEFを注目ベクトルとして設定する。そして、ベクトルEFの midpoint I の座標を計算し（S4715）、入力画像データから、始点、終点、midpointの各座標に対応する位置の色情報を取得し（S4720）、これらの平均値を計算して、ベクトルEFの色情報とする（S4725）。この処理を順次残りのベクトル、すなわち、ベクトルFG（midpoint J）、ベクトルGH（midpoint K）、ベクトルHE（midpoint L）に対して行う。これにより、各ベクトル（ベクトルEF、ベクトルFG、ベクトルGH、ベクトルHE）に色情報を付加することができる（以上、図15（C）参照）。

【0116】一方、閉領域色検出処理では、閉領域と認識されたベクトル群が閉領域色検出部33に入力されると、図6のステップS4750（図8参照）で、すべてのベクトルに対して、ベクトル上の1点からベクトルに垂直な方向に閉領域の内部へ所定の距離、 $\{(線幅/2) + \alpha\}$ ドットだけ進んだ点（内部点）の座標を計算し（S4755）、入力画像データから、内部点の座標に対応する位置の色情報を取得する（S4760）。そして、すべてのベクトルについてのベクトル長さ分の各色情報に対してヒストグラムを作成し（S4780）、その最多値を求めて、当該閉領域の色情報とする（S4785）。これにより、当該閉領域（ベクトル群）に対して閉領域内部の塗り潰し色として色情報を付加することができる（以上、図15（D）参照）。

【0117】したがって、本実施の形態によれば、図形をベクトルデータに変換する際に、線図形を線幅の情報を持つベクトルデータに変換するので、ベクトル変換後のデータ量をより一層低減することができ、また、アプリケーションソフトでデータを再利用する際にそのデータを加工（編集）しやすくなり、データの再利用時における使い勝手の向上を図ることができる。

【0118】また、このように線図形は線幅の情報を持つベクトルデータに変換されるので、枠線を持つ閉領域は1つの塗り潰し閉領域として認識され、データ量の低減やデータ再利用時の使い勝手の向上に加えて、拡大・縮小の編集作業における線幅の変化を防止することができる。

【0119】また、ベクトル自体の色や閉領域内部の色を検出してベクトルデータに付加するので、図形の色（線自体の色や閉領域内部の塗り潰し色）を忠実に再現することができる。

【0120】これらを図面を使ってさらに具体的に説明すると、次のとおりである。

【0121】まず、図16を参照して、黒色の線分をベクトル変換する場合を説明する。図16に示すように、線分は、従来の方法では、4本のベクトルで囲まれた閉領域で内部が黒色で塗り潰されているものとして認識されるが、本発明では、黒色の線分で線幅8（ポイント）を持つ1本のベクトルとして認識される。したがって、線分として描かれる図形は、ベクトル変換後に4本のベクトルで囲まれた閉領域として出力される（従来技術）よりも、線幅の情報を持つ1本のベクトルとして出力される（本発明）ほうが、アプリケーションソフトでデータを再利用する際に加工（編集）しやすく、かつ、ベクトルデータ量も少なくて済む。

【0122】次に、図17を参照して、黒色の枠線を持つ閉領域で内部が緑色で塗り潰されたものをベクトル変換する場合を説明する。

【0123】かかる閉領域は、従来の方法では、2つの閉領域としてベクトル化され、外側の大きい四角形は内部が黒色で塗り潰され、内側の小さい四角形は内部が緑色で塗り潰されているものとして認識されるが、本発明では、1つの閉領域としてベクトル化され、四角形の枠線が黒色でかつ線幅8（ポイント）で内部が緑色で塗り潰されている閉領域として認識される。よって、本発明によるベクトル変換のほうが、アプリケーションソフトでデータを再利用する際に加工（編集）しやすく、かつ、ベクトルデータ量も少なくて済む。

【0124】また、図17に示すように、かかる閉領域をベクトル変換した後に1.5倍に拡大する場合、従来の方法では、かかる閉領域は上記のように2つの閉領域としてベクトル化されてしまうため、枠線の線幅が変わってしまうが、本発明では、上記のように、枠線は線幅の情報を持つベクトルとしてベクトル化され、内部は緑色に塗り潰されていると認識されているので、拡大しても線幅は変化しない（縮小の場合も同様）。

【0125】なお、本実施の形態では、本発明に係る画像処理装置をスタンドアローンのデジタル複写機100に適用した場合（図18参照）を例にとって説明したが、これに限定されるわけではない。たとえば、本発明に係る画像処理装置は、図19に示すように、ネットワーク200に接続された画像入力装置210（たとえば、スキャナ）と画像形成装置220（たとえば、プリンタ）からなる複写システムにも適用可能である。この場合、本発明に係る画像処理装置は、画像処理装置210と画像形成装置220のいずれに搭載してもよい。

【0126】以下、本発明のさらに他の適用例をいくつか簡単に説明しておく。なお、いずれの場合も、たとえば、図1に示す画像処理部7を備えており、画像データ（ラスターデータ）の中の図形領域を線幅や色の情報を持つベクトルデータに変換する機能（高度ベクトル変換機

能）を有している。

【0127】（第2の実施の形態）図20に示す適用例は、ベクトル変換後の図形データ（ベクトルデータ）を印刷（用紙に出力）する代わりに、電子ファイルの状態で出力する場合である。すなわち、図20に示す第2の実施の形態に対応する画像処理装置は、高度ベクトル変換機能を有する電子ファイル作成機300として構成されている。

【0128】電子ファイル作成機300は、図示しないが、スキャナエンジン、上記画像処理部7、各種外部機器（たとえば、各種PDA、ノートパソコンなど）と接続するための各種外部出力ポート、リムーバブルタイプの各種記憶メディア（たとえば、フロッピー（登録商標）ディスク、Zip、PCカード型メモリ（以下単に「PCカード」という）、MOディスクなど）を読み書きするための各種記録メディアドライブ、およびLANなどのネットワークと接続するためのネットワークポートなどを有する。ここでは、外部出力ポート、記録メディアドライブ、ネットワークポートの3つをまとめて「外部出力部」と呼ぶことにする。外部出力ポートは、たとえば、USBやIEEE1394、IEEE1284などのシリアル/パラレルポートで構成され、ネットワークポートは、たとえば、イーサネット（登録商標）（Ethernet（登録商標））などで構成されている。また、記録メディアドライブには、たとえば、フロッピーディスクドライブ、Zipドライブ、PCカードドライブ、MOディスクドライブなどがある。

【0129】スキャナエンジンで読み取って得られた画像データ（ラスターデータ）は、画像処理部7で処理され、その結果、所定の形式（たとえば、PDF）の画像ファイルが作成される（図3参照）。この時、前述のように、画像データの中の図形領域は、線幅や色の情報を持つベクトルデータに変換されている。作成された画像ファイルは、図20に示すように、適宜、外部出力部（外部出力ポート、記録メディアドライブ、ネットワークポート）から所望の出力先（たとえば、パソコン、フロッピーディスク、Zip、PCカード、MOディスク、ネットワークなど）に出力される。

【0130】（第3の実施の形態）図21に示す適用例は、原稿を直接読み取って画像データを取得する代わりに、外部機器（たとえば、外部スキャナ、デジタルカメラ、メモリ内蔵型ハンディスキャナなど）を用いて取得された画像データを取り込む場合である。すなわち、図21に示す第3の実施の形態に対応する画像処理装置は、高度ベクトル変換機能を有する印刷機400として構成されている。

【0131】印刷機400は、図示しないが、各種外部機器（外部スキャナ、デジタルカメラ、メモリ内蔵型ハンディスキャナなど）と接続するための各種外部入力ポート、上記画像処理部7、およびプリンタエンジンなど

を有する。さらに、各種の記録メディアドライブやネットワークポートを有することもできる。この場合、外部入力ポート、記録メディアドライブおよびネットワークポートは、外部から画像データを取り込むために使用される。

【0132】外部機器（外部スキャナ、デジタルカメラ、メモリ内蔵型ハンディスキャナなど）を用いて取得された画像データは、外部入力ポートを通じて取り込まれ、画像処理部7で処理された後、プリンタエンジンで印刷される。すなわち、この場合は、図21に示すように、適宜、所望の画像入力先（外部スキャナ、デジタルカメラ、メモリ内蔵型ハンディスキャナなど）から画像データを取り込んで印刷を行う。

【0133】（第4の実施の形態）図22に示す適用例は、ネットワークを経由して画像データを受信した後、高度ベクトル変換機能を含む画像処理を行い、得られた画像ファイルを再びネットワーク経由で外部に送信する場合である。すなわち、図22に示す第4の実施の形態に対応する画像処理装置は、ネットワーク上で他のコンピュータなどに対して高度ベクトル変換機能を含む画像

処理のサービスを提供するサーバ500（以下「画像処理サーバ」という）として構成されている。

【0134】画像処理サーバ500は、図示しないが、ネットワークポートおよび上記画像処理部7などを有する。この場合、高度ベクトル変換機能に必要なパラメータ（たとえば、しきい値など）は、必ずしも操作パネルから入力する必要はなく、ネットワーク経由で外部から入力される。

【0135】ネットワーク経由でネットワークポートを通じて受信された画像データは、画像処理部7で処理され、その結果、所定の形式（たとえば、PDF）の画像ファイルが作成される（図3参照）。作成された画像ファイルは、図22に示すように、要求に応じて、再びネットワークポートを通じてネットワーク経由で外部に送信される。すなわち、この場合は、図22に示すように、ネットワーク上の任意のコンピュータで動作中のあるアプリケーション（送信元）から画像データと必要なパラメータを受信して、高度ベクトル変換機能を含む画像処理を行い、線幅や色の情報を持つベクトルデータを含んだ画像ファイルを再びネットワーク上の任意のコンピュータ（複数でもよい）で動作中のあるアプリケーション（送信先）に送信する。

【0136】（第5の実施の形態）図23に示す適用例は、本体内に特定のスキャナエンジンやプリンタエンジンを設けることなく、単に高度ベクトル変換機能を含む画像処理機能のみを持たせて、必要に応じて外部入力機器（たとえば、スキャナなど）や外部出力機器（たとえば、プリンタなど）を接続して利用するようにした場合である。すなわち、図23に示す第5の実施の形態に対応する画像処理装置は、高度ベクトル変換機能を有する

コントローラ600として構成されている。

【0137】コントローラ600は、図示しないが、外部入力ポート、上記画像処理部7、および外部出力ポートなどを有する。なお、外部入力ポートおよび外部出力ポートは、それぞれ、データの入力と出力が可能な入出力ポートであってもよい。ここでは、コントローラ600に外部入力機器としてスキャナを、外部出力機器としてプリンタをそれぞれ接続して、当該コントローラ600をスキャナ/プリンタコントローラとして機能させる場合を例示してある。

【0138】スキャナを用いて原稿を読み取って得られた画像データは、外部入力ポートを通じて取り込まれた後、画像処理部7で処理され、その結果、所定の形式（たとえば、PDF）の画像ファイルが作成される（図3参照）。作成された画像ファイルは、図23に示すように、外部出力ポートを通じてプリンタへ送信される。

【0139】なお、上記各実施の形態において、高度ベクトル変換機能は、上記処理手順（図3～図8参照）を記述した所定のプログラムをCPUが実行することによって行われるものであり、この所定のプログラムはコンピュータ読取可能な記録媒体（たとえば、フロッピーディスクやCD-ROMなど）によって提供されることもできる。この場合、コンピュータ読取可能な記録媒体に記録されているプログラムは、通常、ハードディスクに転送され記憶される。また、この所定のプログラムは、たとえば、単独で上記各処理を実行するアプリケーションソフトとして提供されてもよいし、また、各装置100～600の一機能として各装置100～600のソフトウェアに組み込んでもよい。

【0140】

【発明の効果】以上説明したように、本発明によれば、図形をベクトルデータに変換する際に、線図形を線幅の情報を持つベクトルデータに変換するので、ベクトル変換後のデータ量をより一層低減することができ、また、アプリケーションソフトでデータを再利用する際にそのデータを加工（編集）しやすくなり、データの再利用時における使い勝手の向上を図ることができる。

【0141】また、このように線図形は線幅の情報を持つベクトルデータに変換されるので、枠線を持つ閉領域は1つの塗り潰し閉領域として認識され、データ量の低減やデータ再利用時の使い勝手の向上に加えて、拡大・縮小の編集作業における線幅の変化を防止することができる。

【0142】さらに、図形の閉領域内部の色または線図形の色を検出してベクトルデータに付加するので、図形の色を忠実に再現することができる。

【図面の簡単な説明】

【図1】 本発明の第1の実施の形態に対応するデジタル複写機の構成を示すブロック図である。

【図2】 デジタル複写機の動作を示すメインフローチ

ャートである。

【図 3】 図 2 の画像処理の内容を示すフローチャートである。

【図 4】 図 3 のベクトル変換処理の内容を示すフローチャートである。

【図 5】 図 3 の閉領域抽出処理の内容を示すフローチャートである。

【図 6】 図 3 の色検出処理の内容を示すフローチャートである。

【図 7】 図 6 のベクトル色検出処理の内容を示すフローチャートである。

【図 8】 図 6 の閉領域色検出処理の内容を示すフローチャートである。

【図 9】 ベクトル化されたデータ（ベクトル群）の一例を示す図である。

【図 10】 スタックに保存されているデータを示す図である。

【図 11】 スタックに保存されているデータを示す図である。

【図 12】 スタックに保存されているデータを示す図である。

【図 13】 スタックに保存されているデータを示す図である。

【図 14】 スタックに保存されているデータを示す図である。

【図 15】 枠線を持つ閉領域に対する高度ベクトル変換処理を説明するための図である。

【図 16】 線分に対するベクトル変換処理の結果を従来技術と本発明とで比較して示した図である。

【図 17】 枠線を持つ閉領域に対するベクトル変換処理の結果を従来技術と本発明とで比較して示した図である。

【図 18】 第 1 の実施の形態の適用例を示す図である。

【図 19】 第 1 の実施の形態の一変更例を示す図である。

【図 20】 本発明の第 2 の実施の形態に対応する適用例（電子ファイル作成機）を示す図である。

【図 21】 本発明の第 3 の実施の形態に対応する適用例（印刷機）を示す図である。

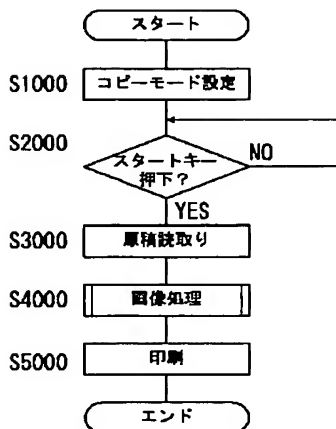
【図 22】 本発明の第 4 の実施の形態に対応する適用例（画像処理サーバ）を示す図である。

【図 23】 本発明の第 5 の実施の形態に対応する適用例（コントローラ）を示す図である。

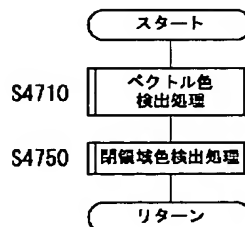
【符号の説明】

7…画像処理部、
19…領域判別部、
21…文字認識部、
23…ベクトル変換部、
25…線幅検出部、
27…閉領域抽出部、
29…色検出部、
31…ベクトル色検出部、
33…閉領域色検出部、
35…コピー画像処理部、
37…合成処理部、
100…デジタル複写機、
200…ネットワーク、
210…画像入力装置、
220…画像形成装置、
300…電子ファイル作成機、
400…印刷機、
500…画像処理サーバ、
600…コントローラ。

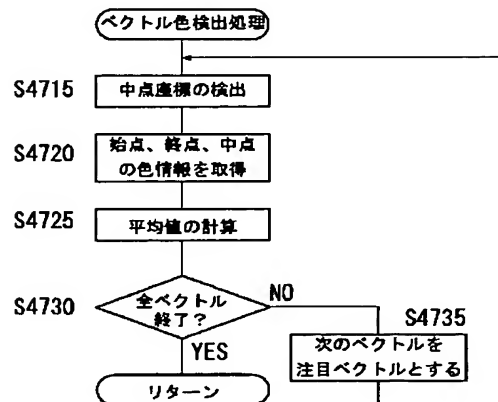
【図 2】



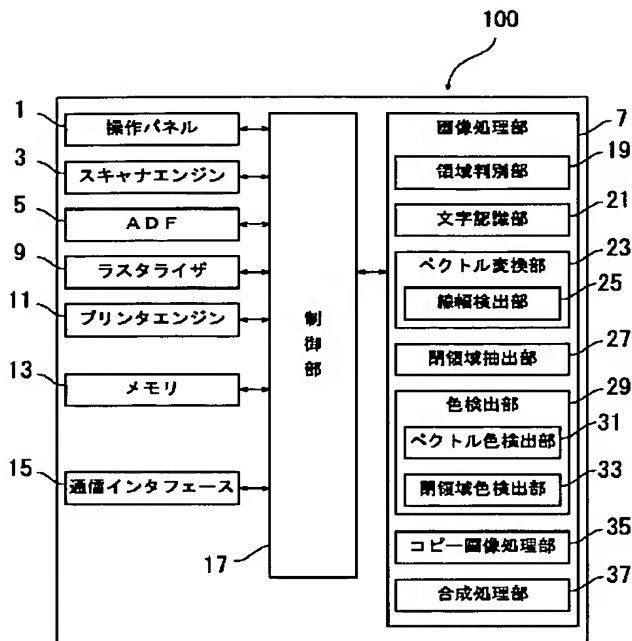
【図 6】



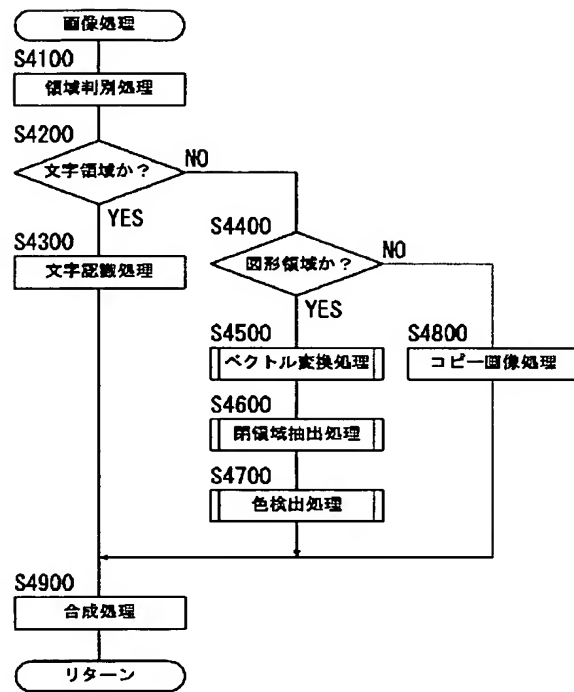
【図 7】



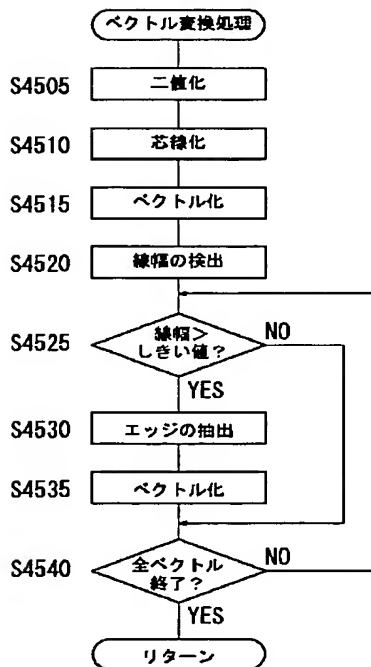
【図1】



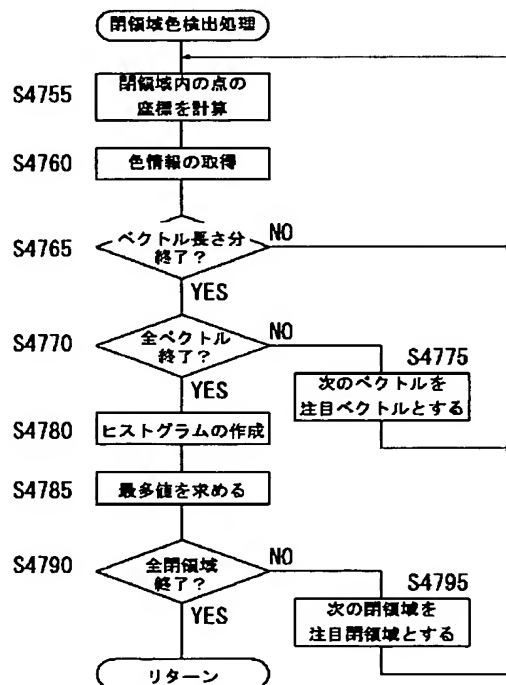
【図3】



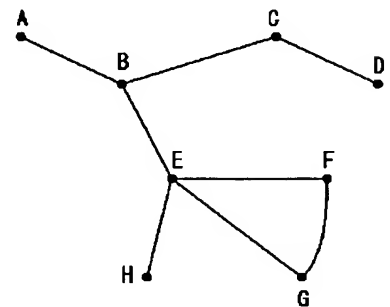
【図4】



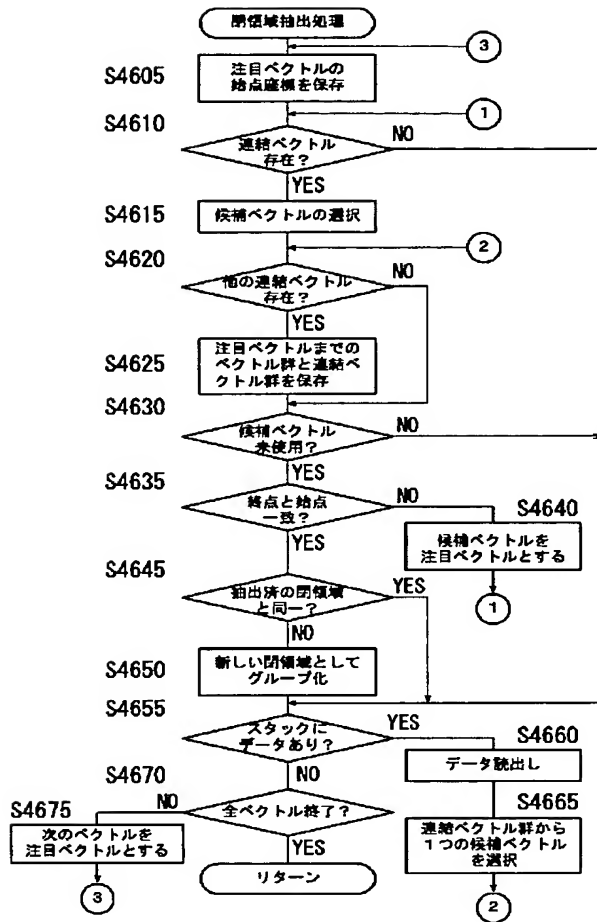
【図8】



【図9】



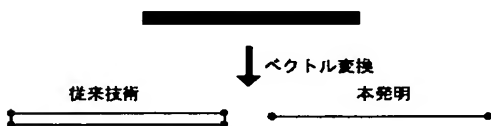
【図 5】



【図 11】

スタック		
	注目ベクトルまでのベクトル群	連結ベクトル群
1	$\overrightarrow{AB}, \overrightarrow{BE}$	$\overrightarrow{EG}, \overrightarrow{EH}$
2		
3		

【図 16】



【図 10】

スタック		
	注目ベクトルまでのベクトル群	連結ベクトル群
1	\overrightarrow{AB}	\overrightarrow{BE}
2		
3		

【図 12】

スタック		
	注目ベクトルまでのベクトル群	連結ベクトル群
1	$\overrightarrow{AB}, \overrightarrow{BE}$	$\overrightarrow{EG}, \overrightarrow{EH}$
2	$\overrightarrow{AB}, \overrightarrow{BE}, \overrightarrow{EF}, \overrightarrow{FG}, \overrightarrow{GE}$	$\overrightarrow{EF}, \overrightarrow{EH}$
3		

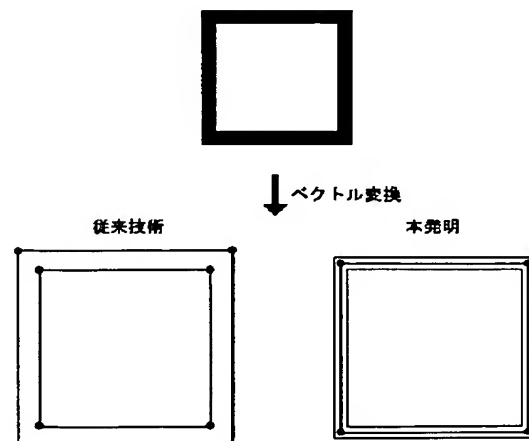
【図 13】

スタック		
	注目ベクトルまでのベクトル群	連結ベクトル群
1	$\overrightarrow{AB}, \overrightarrow{BE}$	$\overrightarrow{EG}, \overrightarrow{EH}$
2	$\overrightarrow{AB}, \overrightarrow{BE}, \overrightarrow{EF}, \overrightarrow{FG}, \overrightarrow{GE}$	\overrightarrow{EH}
3		

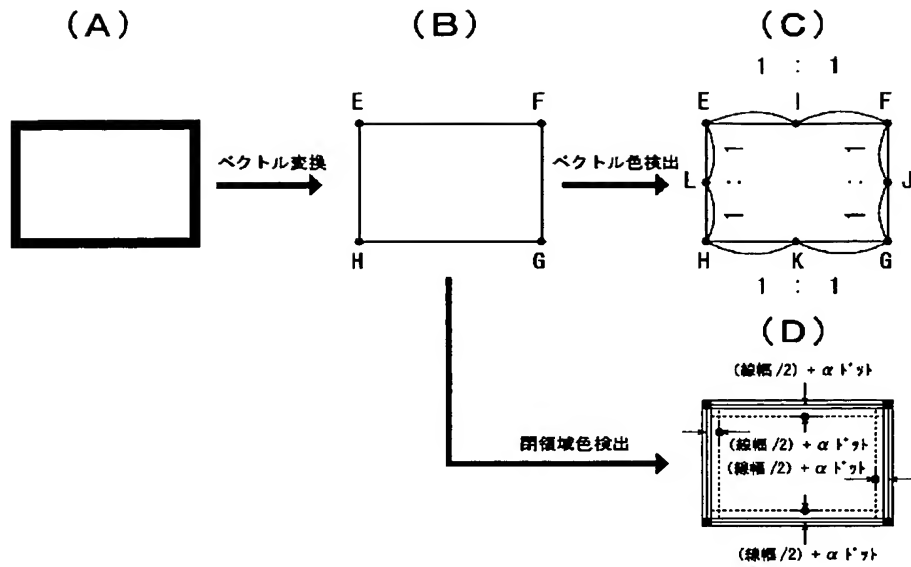
【図 14】

スタック		
	注目ベクトルまでのベクトル群	連結ベクトル群
1	$\overrightarrow{AB}, \overrightarrow{BE}$	\overrightarrow{EH}
2		
3		

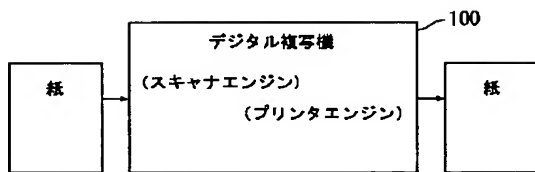
【図 17】



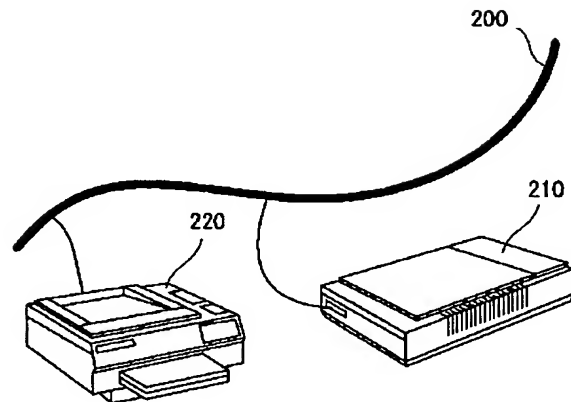
【図15】



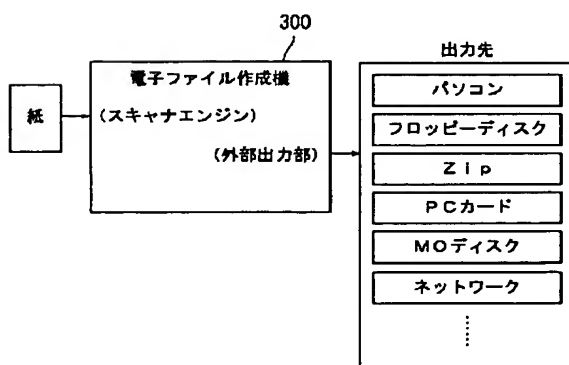
【図18】



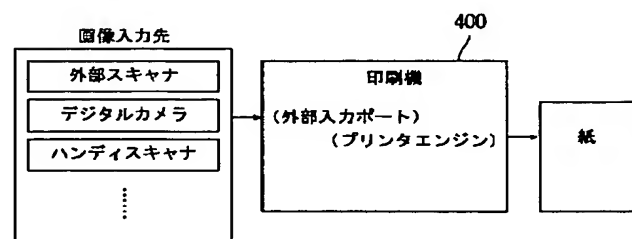
【図19】



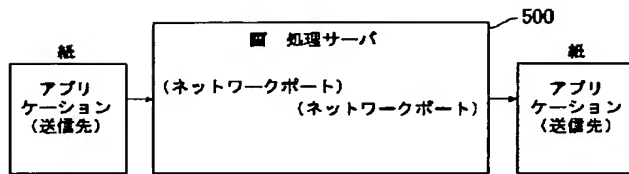
【図20】



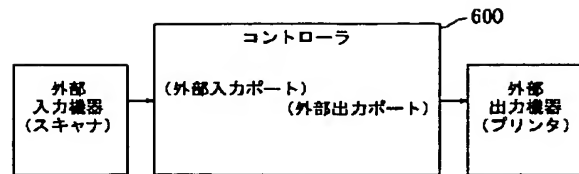
【図21】



【図 22】



【図 23】



フロントページの続き

F ターム (参考) 5B057 BA02 CA01 CA08 CA12 CA16
 CB01 CB08 CB12 CB16 CC04
 CE12 CE15 CF01 DA17 DB02
 DB06 DB09 DC03 DC19 DC25
 5L096 AA02 EA23 EA43 FA06 FA32
 FA35 FA73 GA51